

UNIVERSIDAD
NACIONAL
DE MORENO



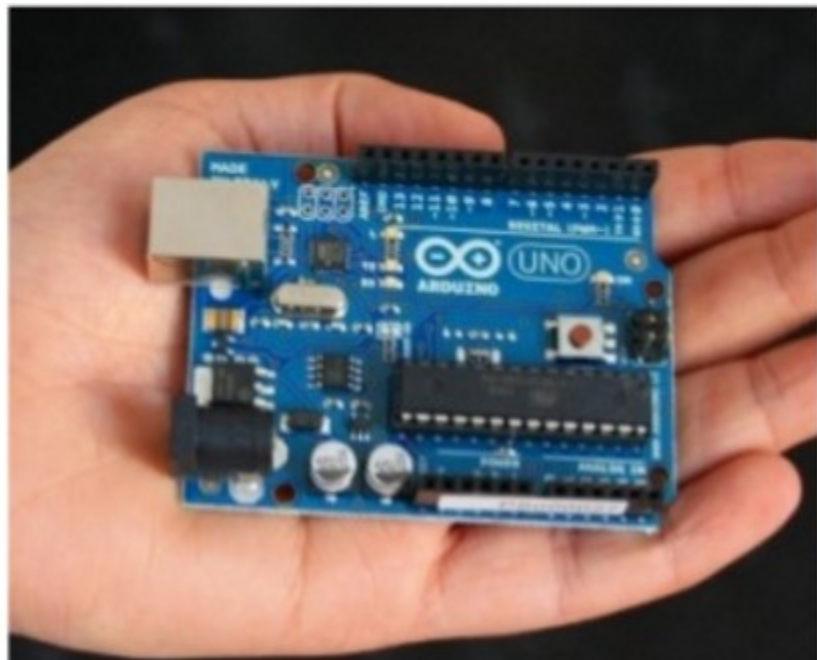
FISICA 1 - INEL

INTRODUCCIÓN A SENSORES CON ARDUINO

Presentación de Prácticas Finales

¿Qué es Arduino?

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en hardware flexible y fácil de usar.



Está Basado en una tarjeta con un microcontrolador que permite conectar sensores, actuadores y otros elementos mediante sus entradas y salidas, analógicas y digitales.

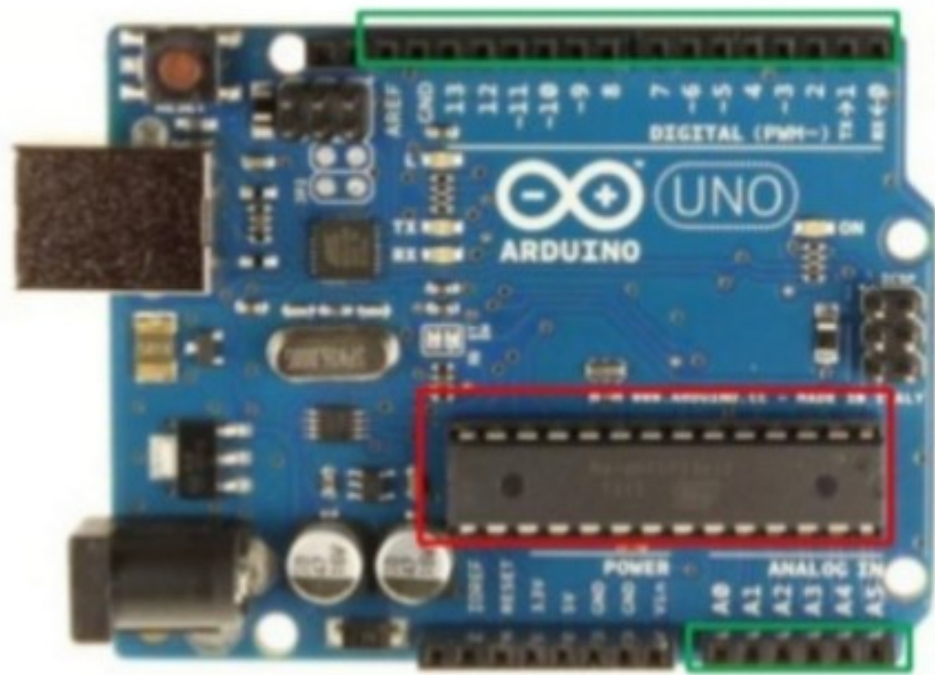


Imagen tomada de: <http://arduino.cc/es/>

Arduino Uno R3



open source
hardware



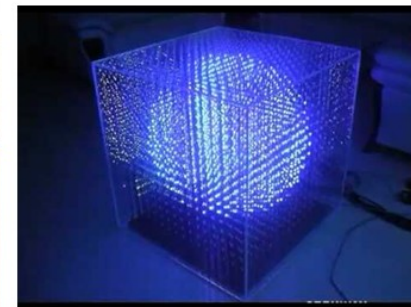
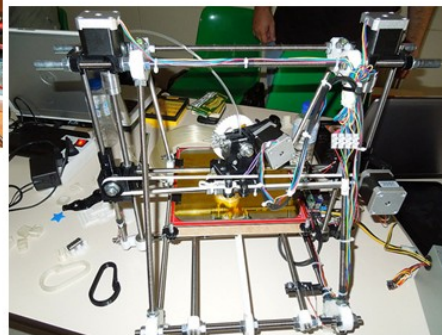
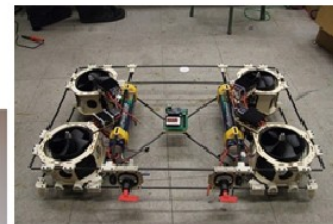
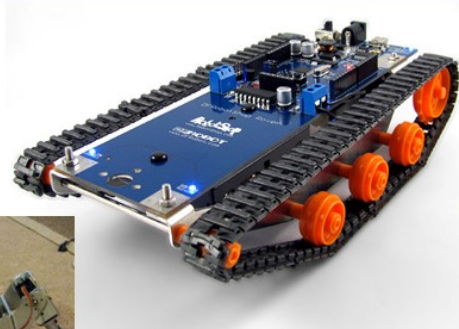
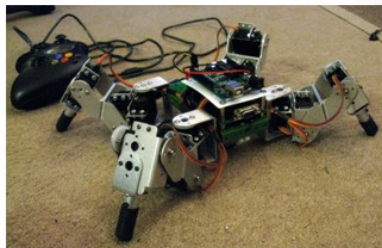
Al ser **Open-Hardware**, tanto su diseño, como su distribución es libre.

Es decir, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin necesidad de adquirir ninguna licencia.



¿Qué distingue a Arduino?

- La filosofía OpenSource – Código Abierto-que lo sustenta
- La comunidad formada a su alrededor
- La sencillez del lenguaje de programación
- El hardware de bajo costo



Tipos de tarjeta Arduino



Due



Mega



Uno



Nano



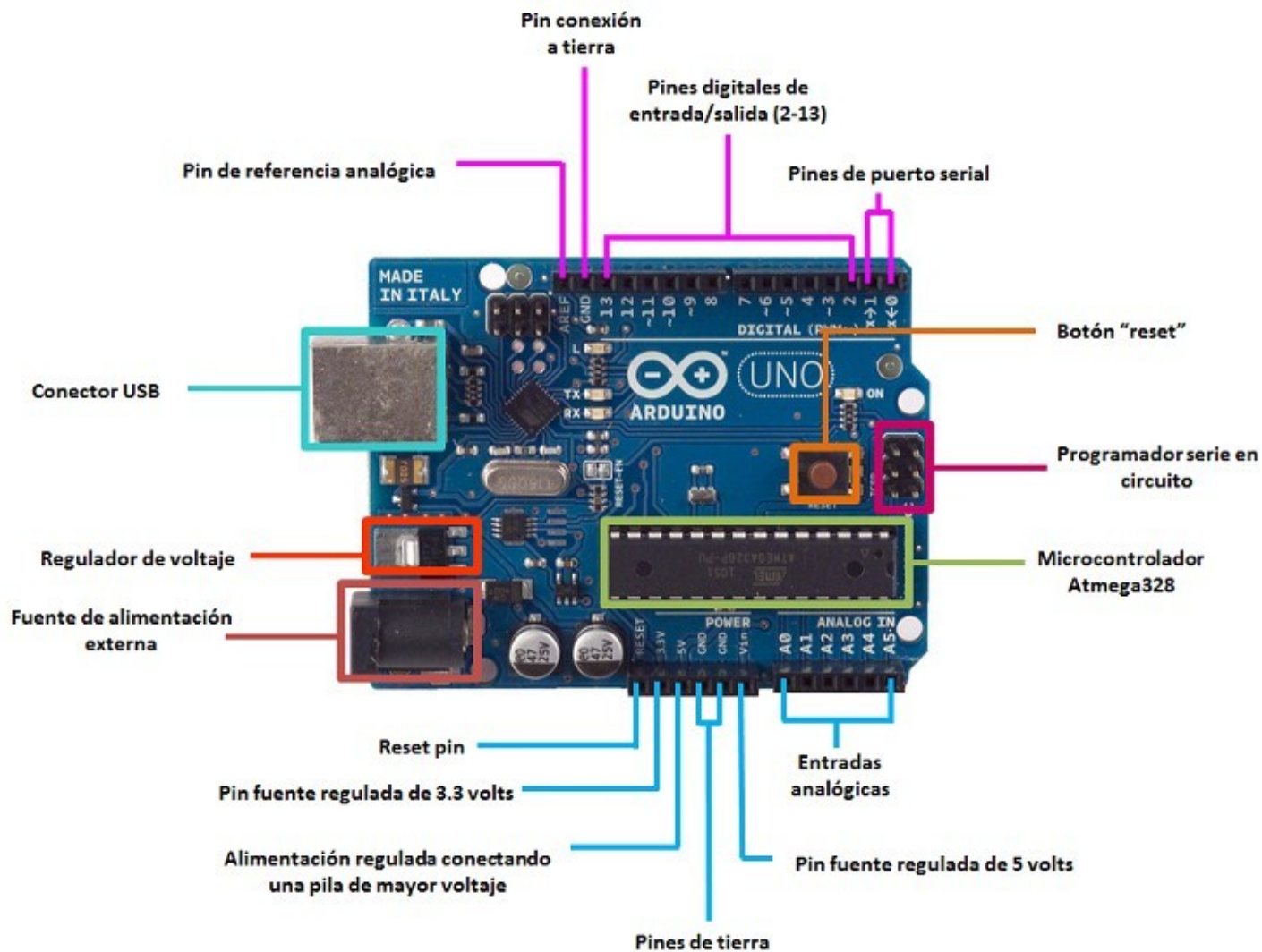
Lilypad



Mini

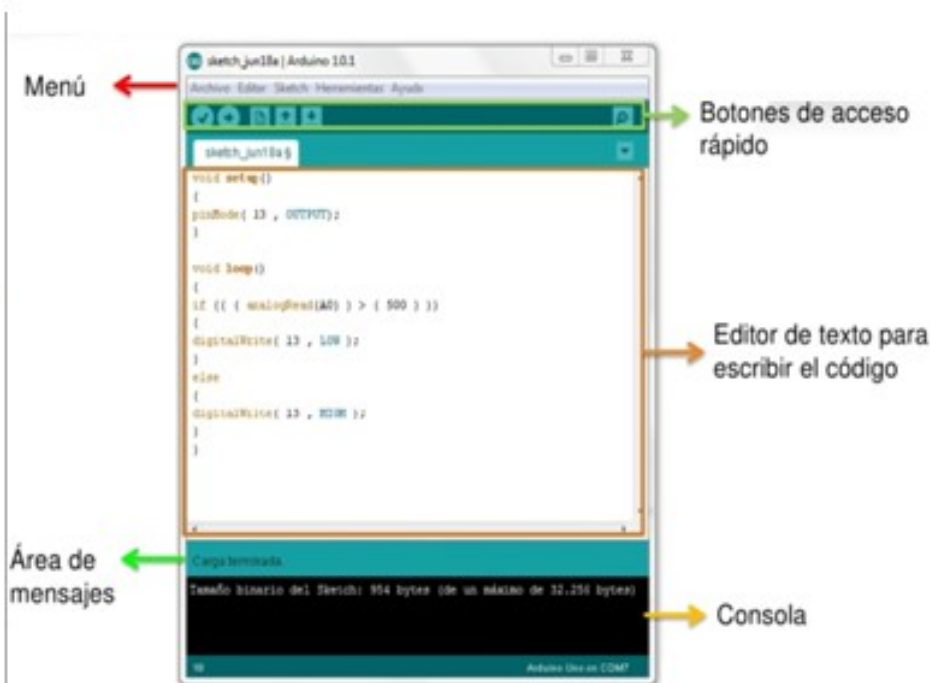


Leonardo



Como se programa Arduino

Arduino cuenta con un entorno de desarrollo nativo creado en Java, por lo que es multiplataforma.



El lenguaje que utiliza es propio de Arduino y está basado en **C**.

Web Arduino:

www.arduino.cc

Manual en castellano 1

Manual en castellano 2



SIGN IN



Language Reference

Arduino programs can be divided in three main parts: *structure*, *values* (variables and constants), and *functions*.

[INGRESAR A LAS REFERENCIAS](#)

Structure

- `setup()`
- `loop()`

Control Structures

- `if`
- `if...else`
- `for`
- `switch case`
- `while`

Variables

Constants

- `HIGH` | `LOW`
- `INPUT` | `OUTPUT` | `INPUT_PULLUP`
- `LED_BUILTIN`
- `true` | `false`
- integer constants
- floating point constants

Data Types

Functions

Digital I/O

- `pinMode()`
- `digitalWrite()`
- `digitalRead()`

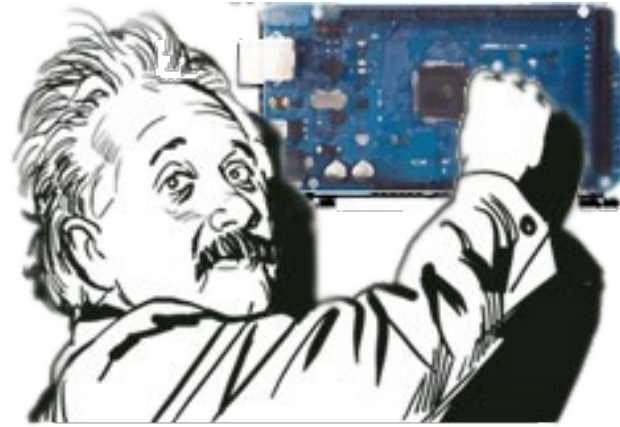
Analog I/O

- `analogReference()`
- `analogRead()`
- `analogWrite()` - *PWM*

UNIVERSIDAD
NACIONAL
DE MORENO



Actividad:



Comenzar a programar Arduino

Sensores para prácticas Finales

Materiales:

Arduino Uno

Cable USB 2.0

Protoboard(Placa de experimentación)

Potenciómetro

LDR

LED

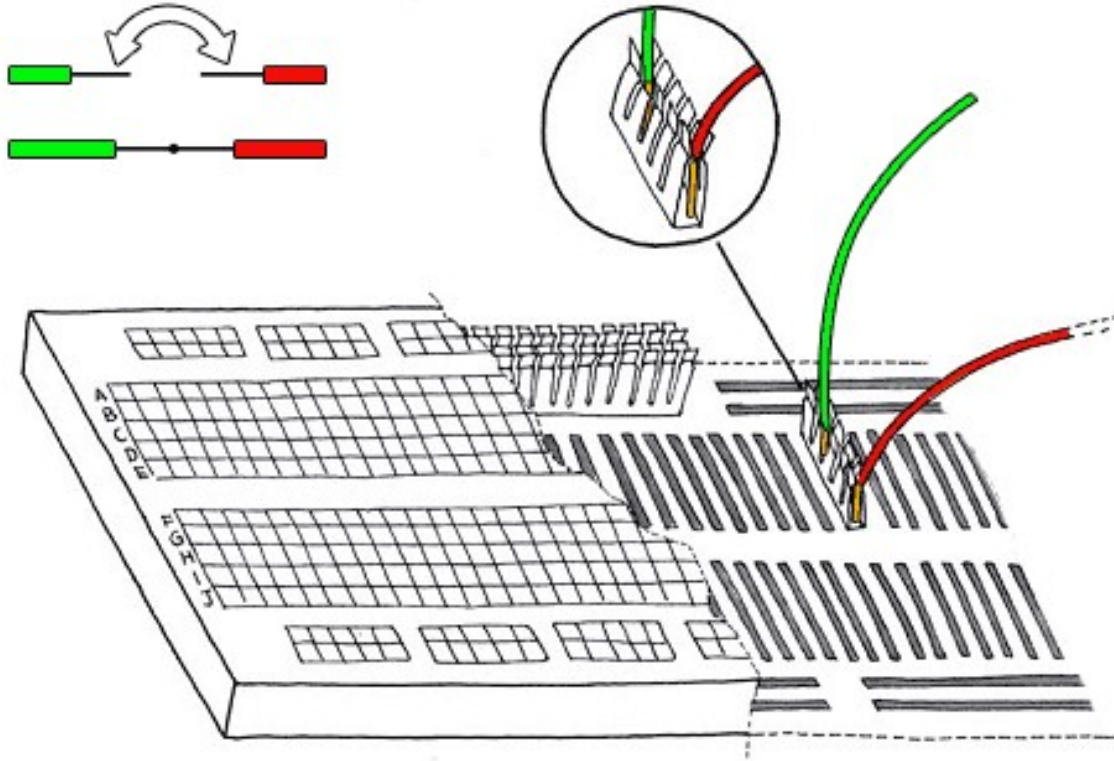
Cables x 7

Sensor Ultrasónico

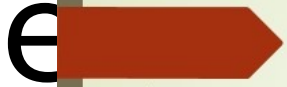
Resistencia (entre 33 ó 330 Ohm)

Pinza

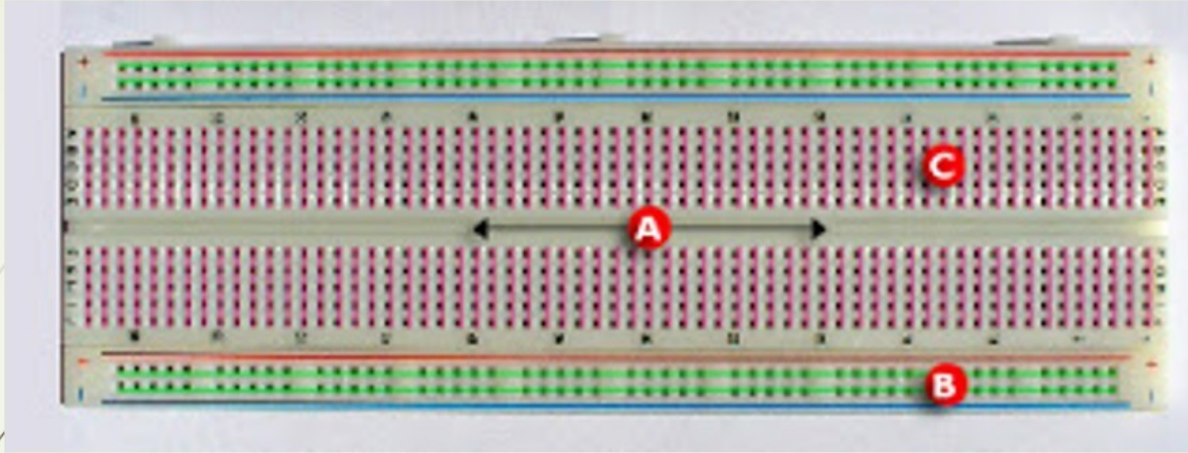
Placa de experimentación:



Placa de

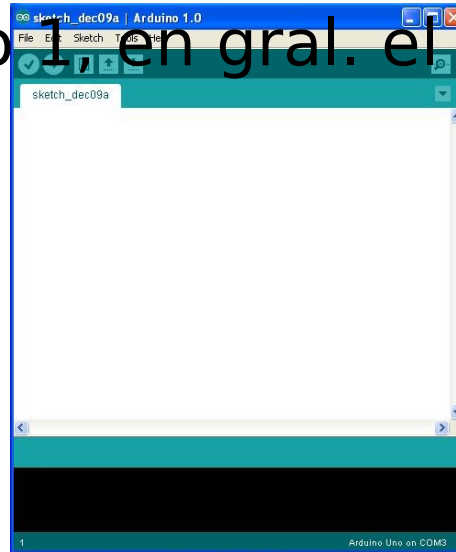


Estructura protoboard

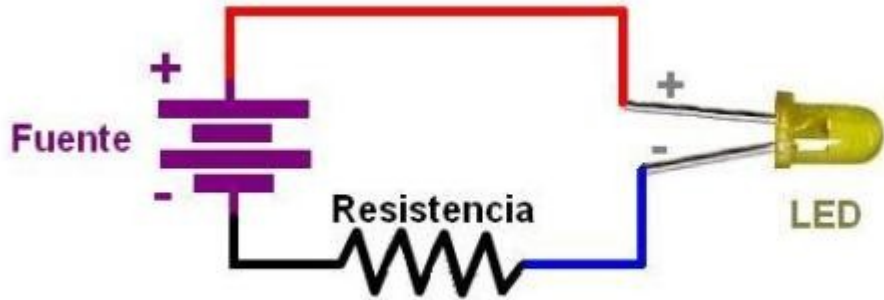


- **A) Canal central:** Es la región localizada en el medio del **protoboard**, se utiliza para colocar los circuitos integrados.
- B) Buses:** Los buses se localizan en ambos extremos del protoboard, se representan por las líneas rojas (buses positivos o de voltaje) y azules (buses negativos o de tierra) y conducen de acuerdo a estas, no existe conexión física entre ellas. La fuente de poder generalmente se conecta aquí.
- C) Pistas:** La pistas se localizan en la parte central del **protoboard**, se representan y conducen según las líneas rosas.

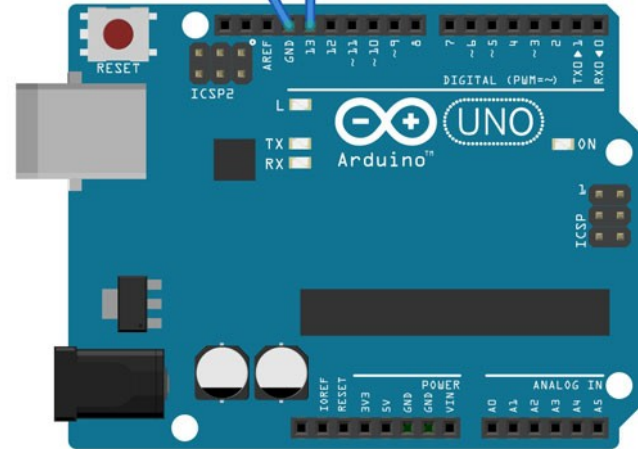
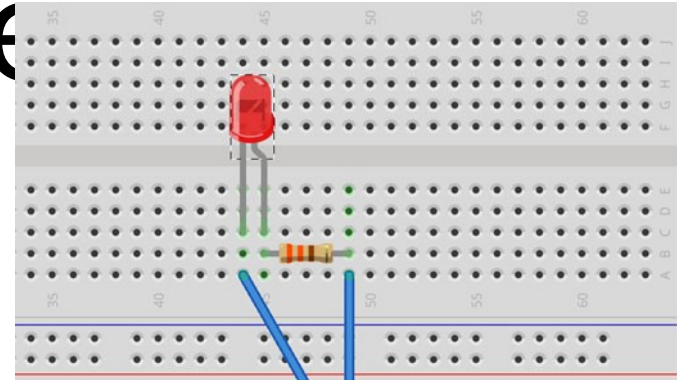
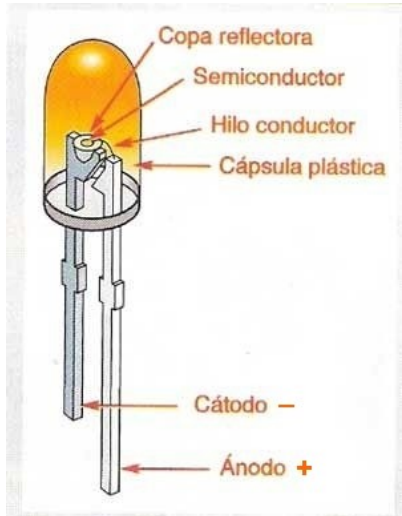
1. Conectar Arduino a PC(ver el número de COM)
2. Abrir Entorno de programación Arduino
3. Ir a Herramientas/ Placa/ seleccionar “Arduino One”
4. Ir a Herramientas/ Puerto/ seleccionar el número de COM(en el paso 1, en gral. el n° más alto)



Circuito Simple

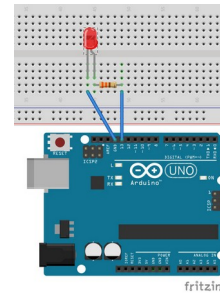


www.afinidadeelectronica.com



fritzing

Parpadeo:
Descargar sketch del Campus
Guardar en una carpeta de tu PC
Abrir Parpadeo.ino



```
/*  
  Enciende un LED durante un segundo, y después apagado por un segundo, repetidamente.  
  */
```

```
void setup()  
  pinMode(13  
}
```

```
void loop()  
  digitalWrite  
  delay(1000  
  digitalWrite  
  delay(1000,  
}
```

MODIFICAR EL CÓDIGO PARA:

ACTIVIDAD N°1: HACER QUE EL LED ESTÉ MÁS TIEMPO APAGADO QUE PRENDIDO

ACTIVIDAD N°2: HACER QUE HAGA DOS DESTELLOS CADA UNO EN TIEMPOS DIFERENTES EN FORMA PERIÓDICA.

```
  // Espera un segundo, osea 1000 milisegundos
```

Entrada Analógica:

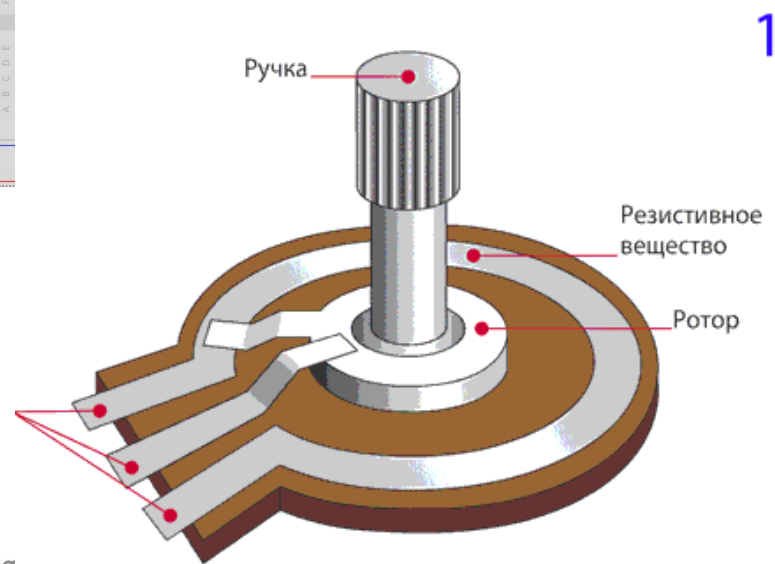
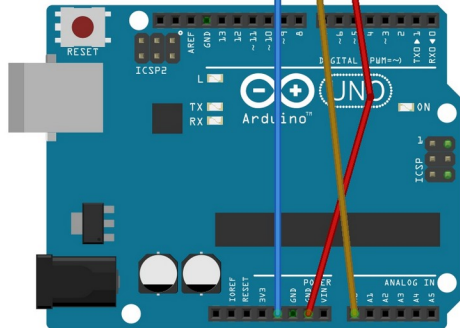
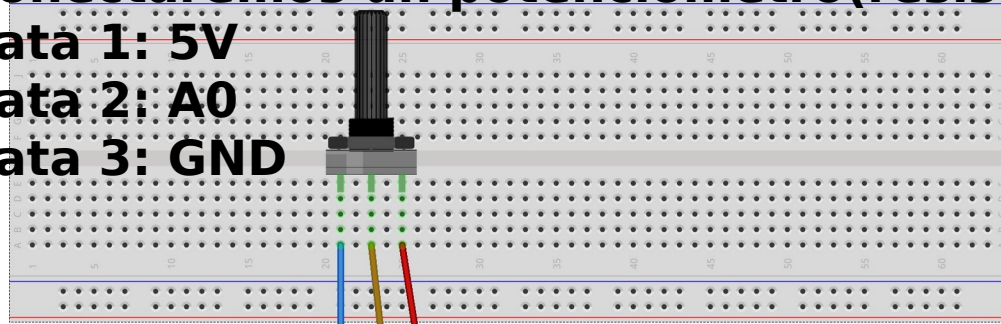
Veremos como Arduino puede recibir información por entrada analógica

Conectaremos un potenciómetro(resistencia variable)

Pata 1: 5V

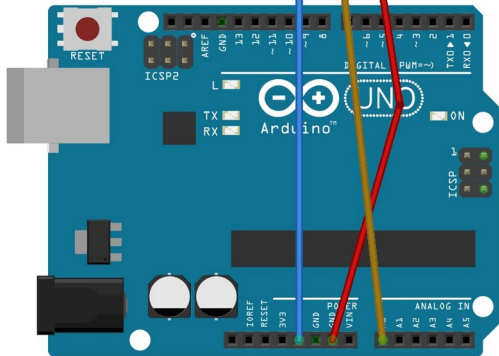
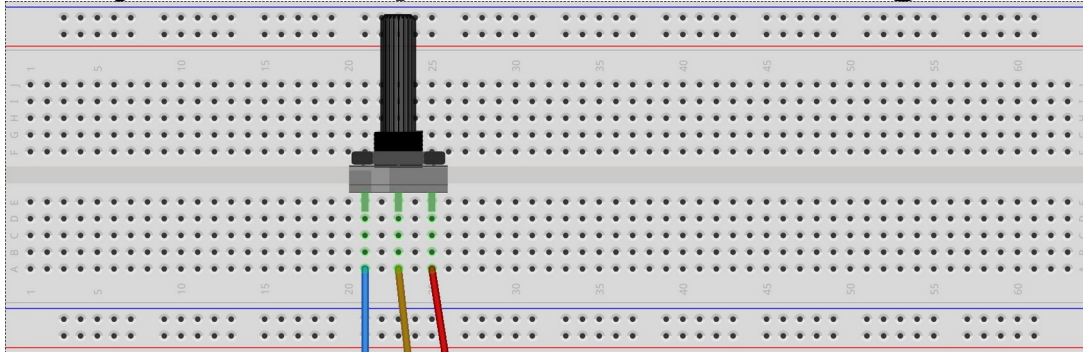
Pata 2: A0

Pata 3: GND



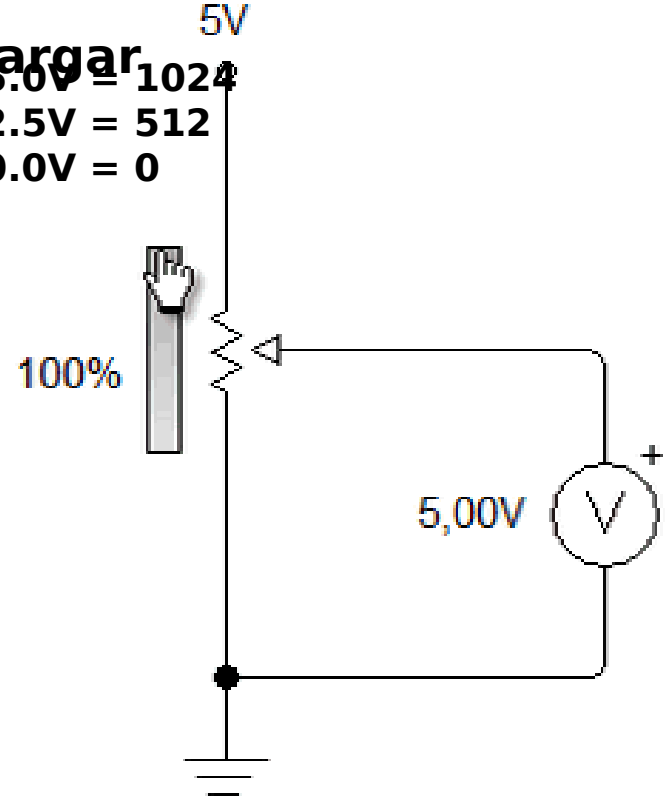
Entrada Analógica:

Ejecutar ImprimirEntradAnalogica.ino y Cargar

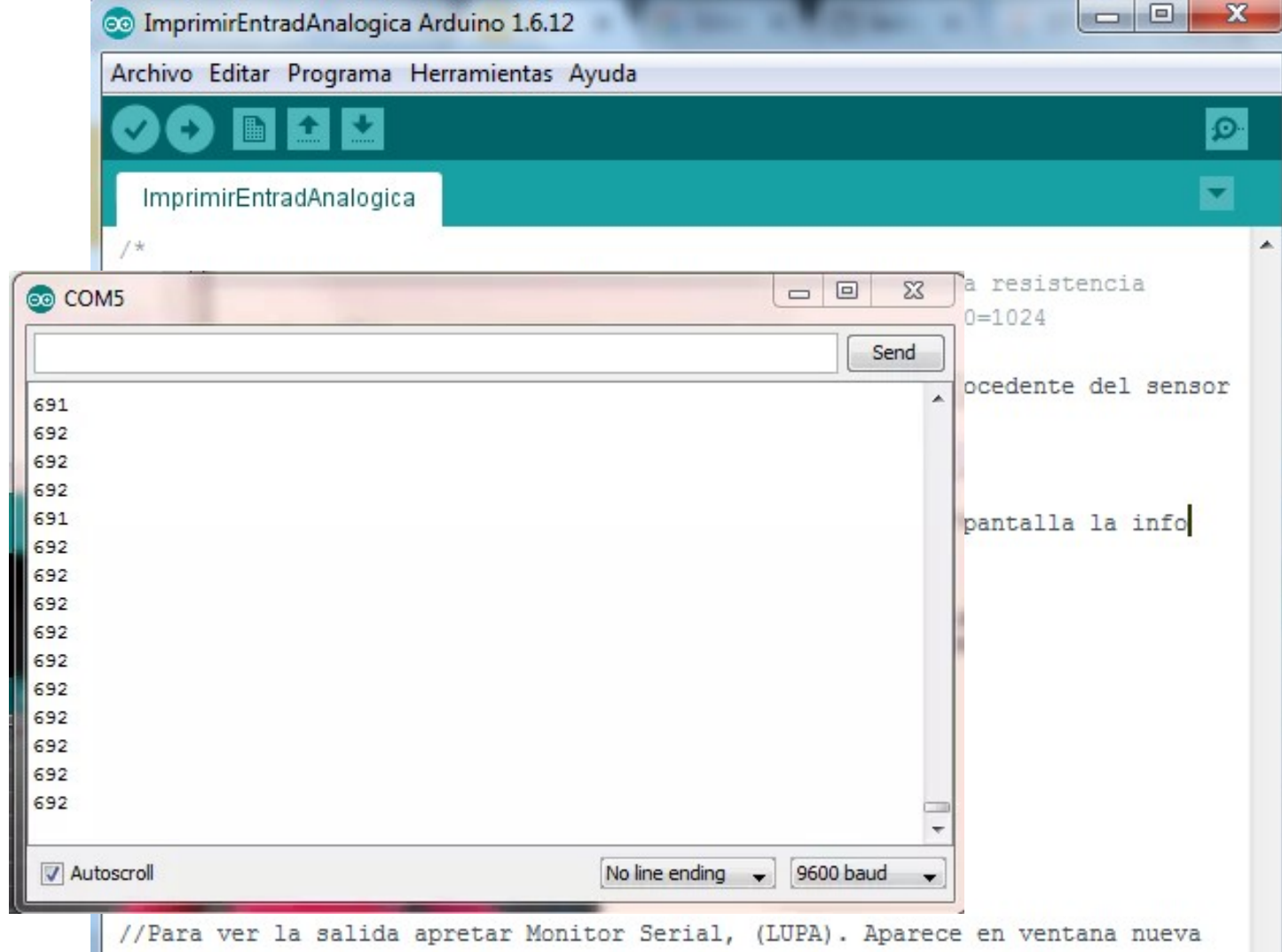


Arduino lee en 10bit
=1024

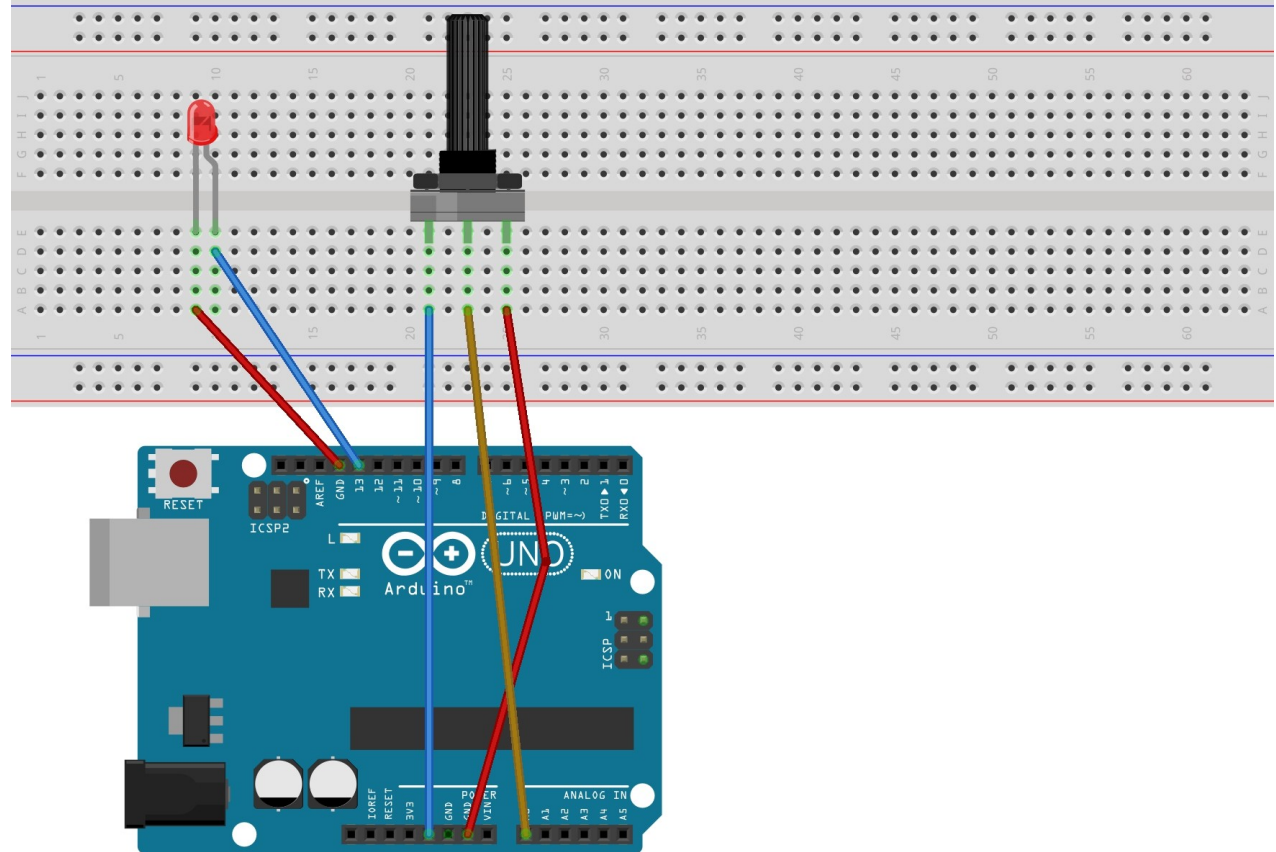
5.0V = 1024
2.5V = 512
0.0V = 0



Imprimir
caída de
Tensión
entre 0 y
1024



Modificar el parpadeo de un Led con el valor del Potenci3́metro



1. Cerrar todos los Sckeths

Abrir ParpadeoRVariable.ino

Cargar

Girar pote

```
ParpadeoRVariable.ino
Usar el potenciómetro para prender y parpadear un LED,
*/
int ValorPote = 0; // variable para almacenar el valor procedente del sensor
int pin=0;
int LED=13;

void setup() {
  Serial.begin(9600); // Iniciamos el puerto USB para leer en pantalla la info
  pinMode(pin, INPUT); // se elige el pin nro A0 como entrada
  pinMode(LED, OUTPUT); // inicializa el pin digital 13(LED) como una salida.
}

void loop() {
  ValorPote = analogRead(pin); // lee el valor del sensor:
  digitalWrite(LED, HIGH); // enciende el LED. HIGH es el nivel alto de voltaje(5volts)
  delay(ValorPote); // Espera lo que marque el potenciómetro(máximo 1024mseg)
  digitalWrite(LED, LOW); // apaga el LED haciendo que el voltaje sea cero.
  delay(ValorPote); // Espera lo que marque el potenciómetro(máximo 1024mseg)
  Serial.print(ValorPote); // Imprimir el valor del sensor
  Serial.println(); // Bajar una línea(enter)
}

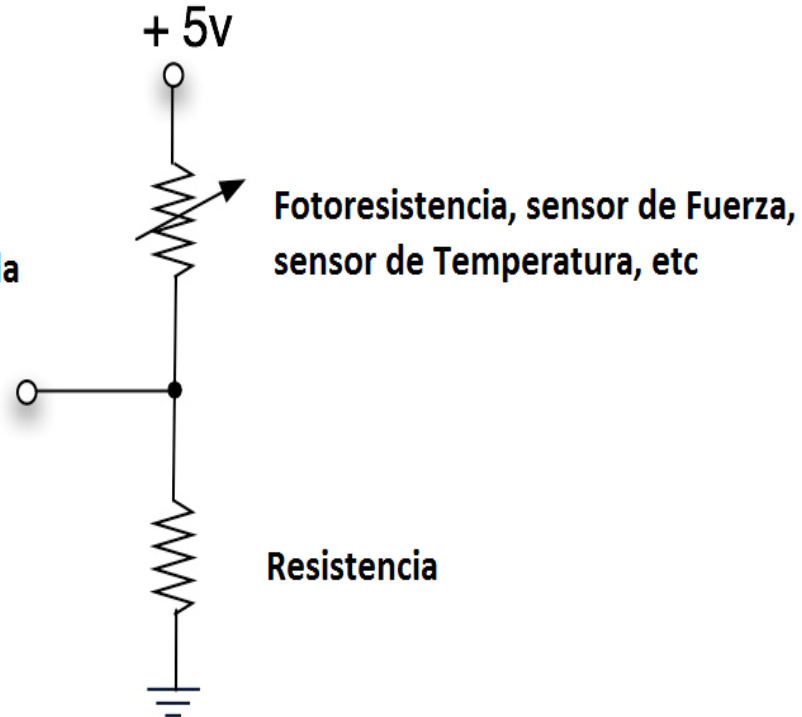
// Cargar código a Arduino con "Subir" ---->
// Girar pote y ver LED
// Para ver la salida apretar Monitor Serial, (Opcional).
```

Hay unas decenas de sensores resistivos, todos se conectan igual.

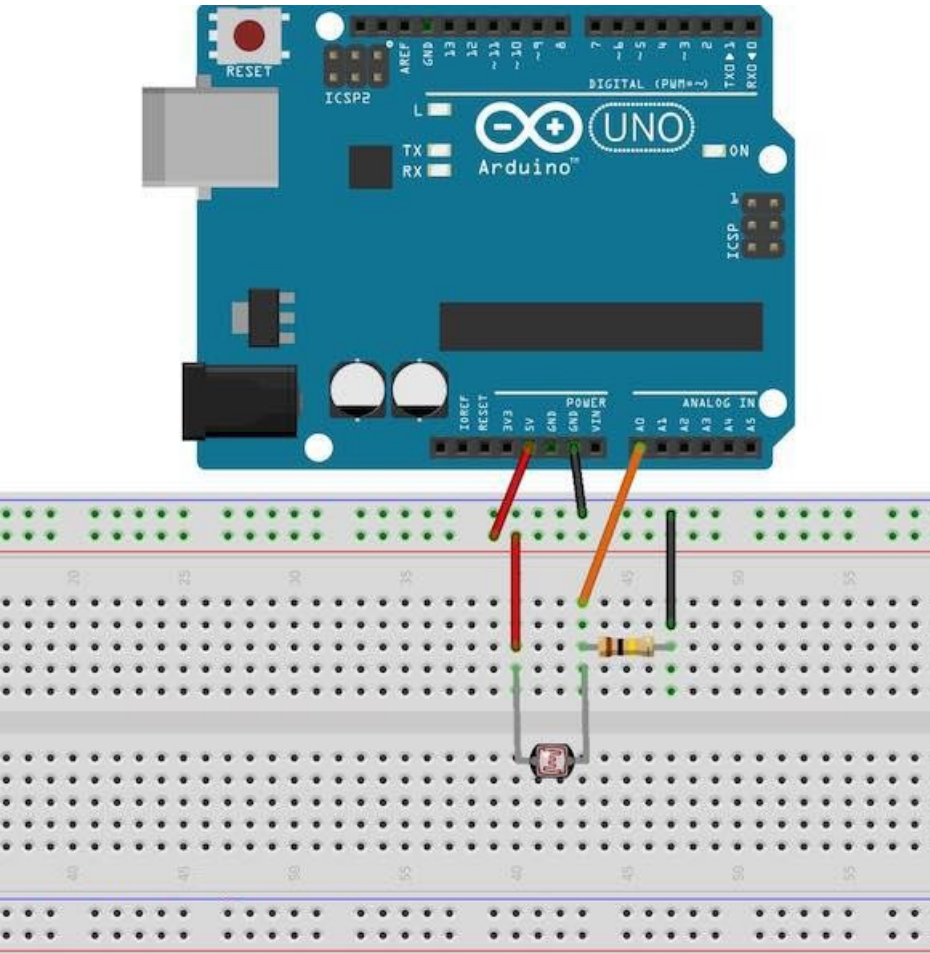
Resistencia Variable conectada a la Entrada Analógica del Arduino

Variable a medir	Sensor resistivo
Mecánica	Potenciómetros y galgas extensométricas
Térmica	Termorresistencia y termistores
Magnética	Magnetorresistencia
Óptica	Fotorresistencia
Química	Higrómetro resistivo

Pin de Entrada Analógica



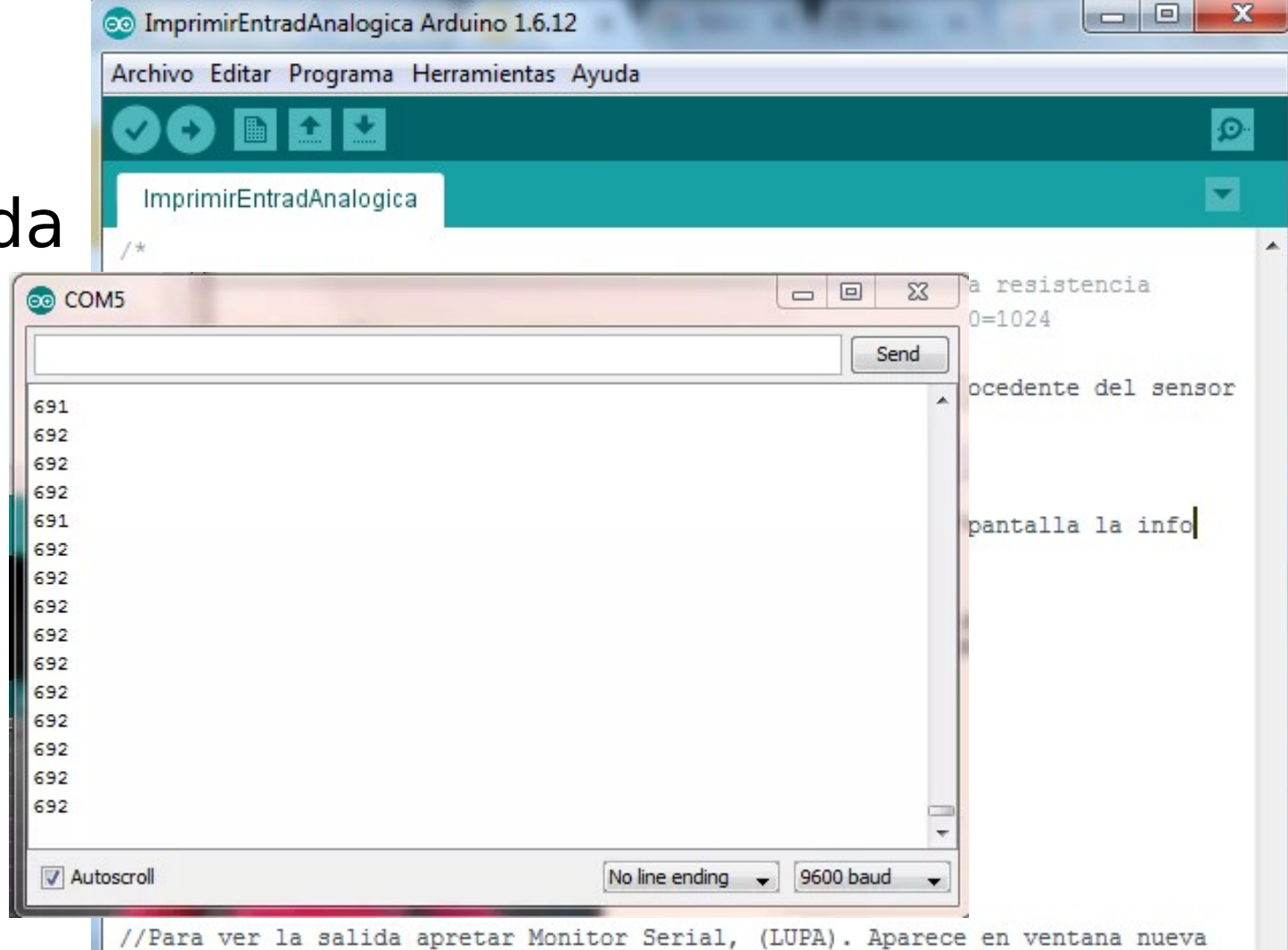
Conexión de Resistencia Dependiente de la Luz



**Cambiar el
Potenciómetro
por el LDR y resistencia
para regular lámpara
LED**

Una LDR es una resistencia con una característica muy particular: su valor depende de la intensidad de luz que incide en ella. Cuando la intensidad de la luz aumenta, el valor de la resistencia descende, y viceversa. Una aplicación es la automatización de los sistemas de iluminación, de tal manera que al oscurecer se enciendan las luces.

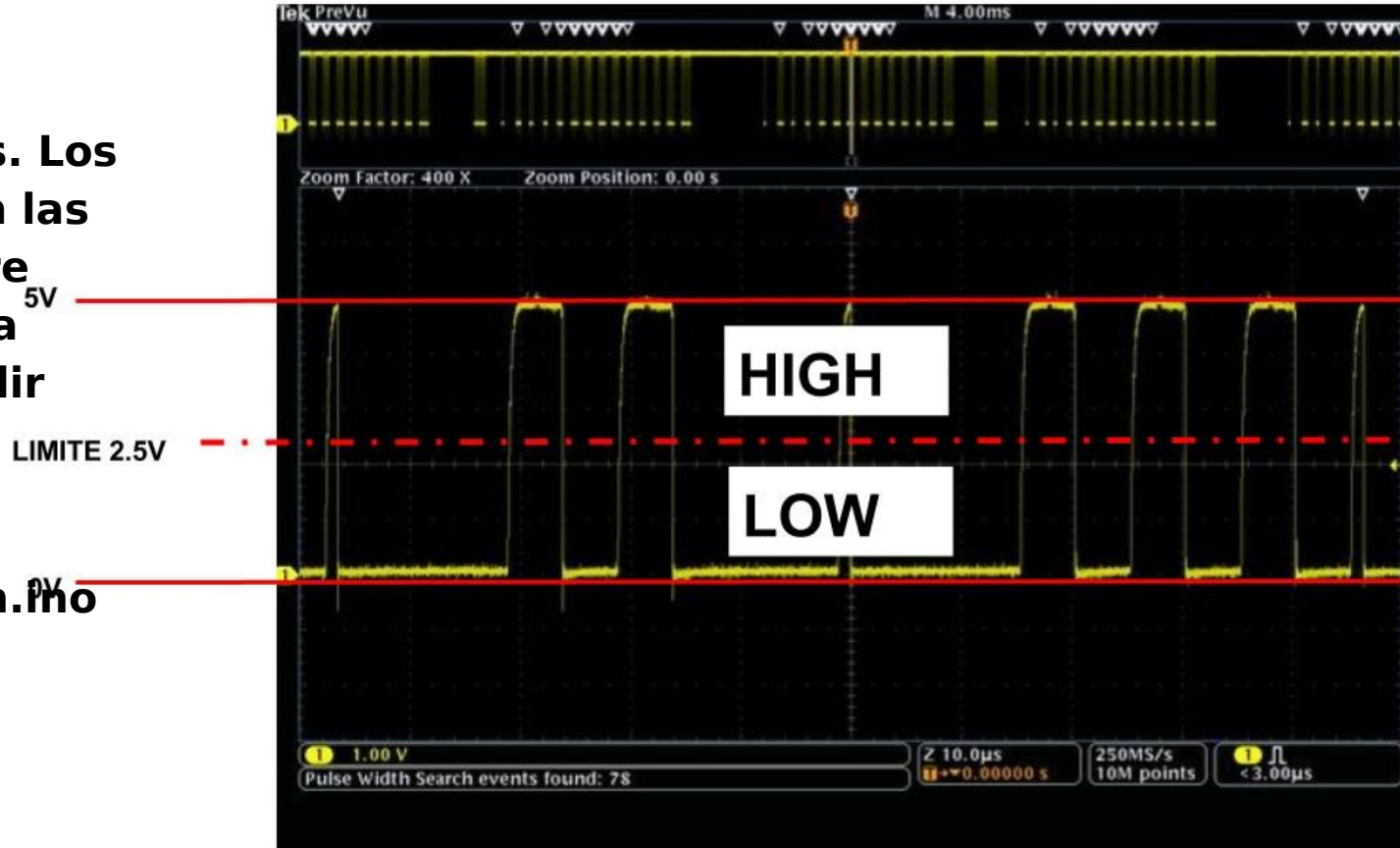
Imprimir una Entrada



Entradas Digitales

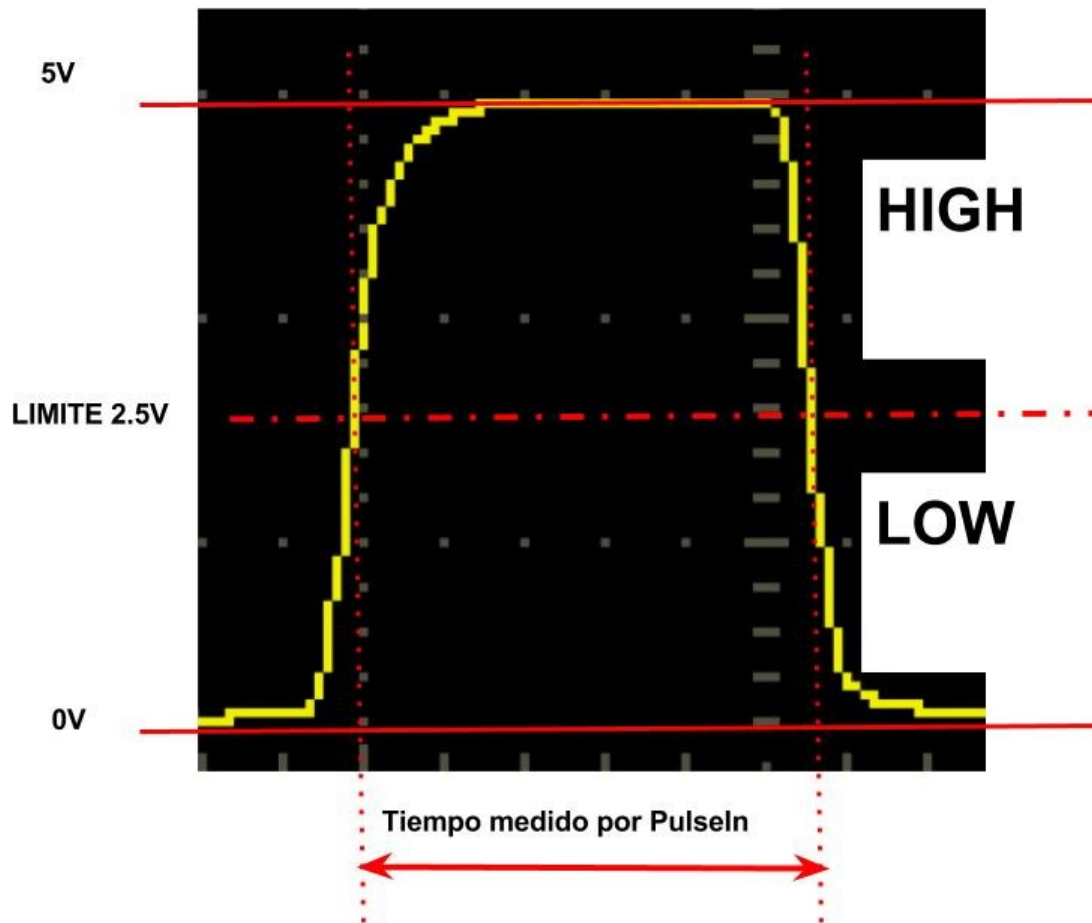
Vamos a usar las compuertas ópticas. Los pulsos que generan las interrupciones entre emisor y receptor la usaremos para medir tiempo

Ejecutar
TiempoInterrupcion.ino



Función PulseIn

Mide en el tiempo que la
señal está por
encima(HIGH) o por
debajo(LOW) de 2.5V



Mediremos el tiempo que se se interrumpe la señal entre emisor y receptor infrarrojo

CONEXION DE LABODUINO A LA COMPUERTA OPTICA

EMISOR INFRARROJO

CONEXION:
AMARILLO ó NARANJA: GND
VIOLETA ó AZUL: 5V
ROSA: PIN 8

RECEPTOR INFRARROJO

TempoInterrupcion Arduino 1.6.12

Archivo Editar Programa Herramientas Ayuda



Tiempo

```
/*  
 * Me  
Conta  
Rosa  
Azul  
Amari
```

```
*/  
  
float  
int p
```

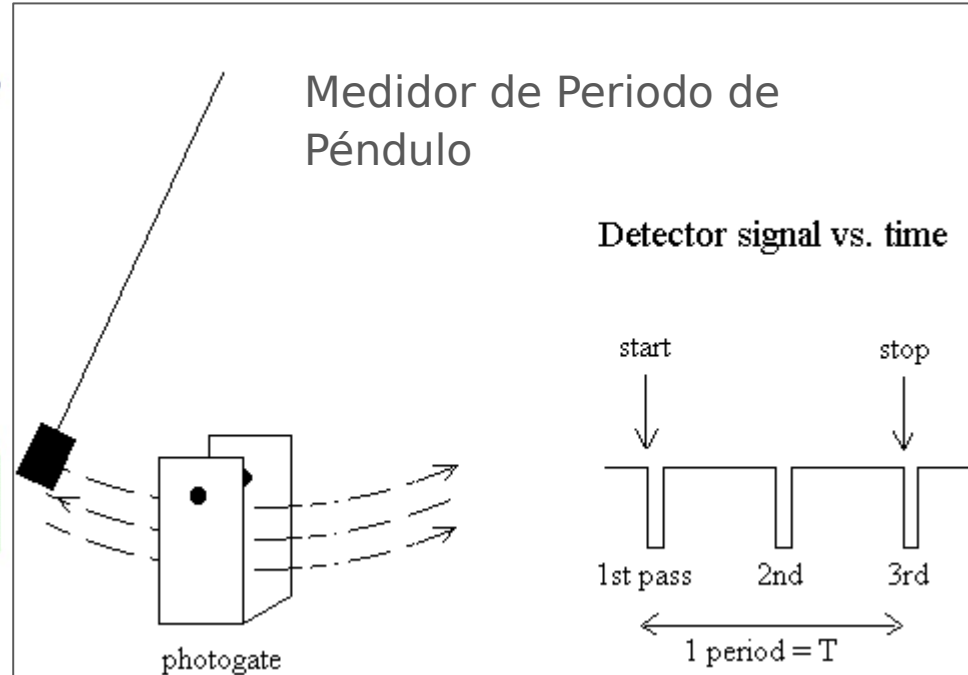
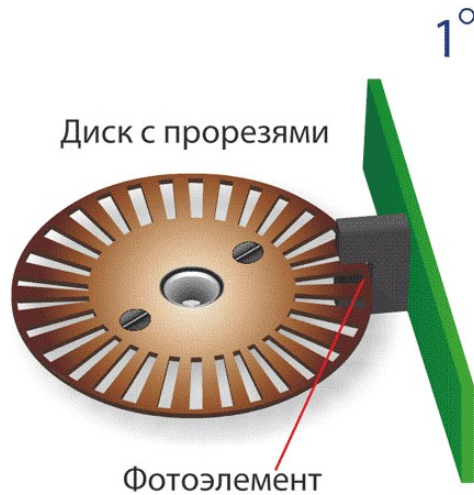
```
void  
//  
Ser  
pi  
}  
void  
//  
Laps  
//I  
Ser  
//b  
Ser  
}  
}
```

Actividad N°1: Diseñar un sketch que mida la velocidad de tu celular en MKS

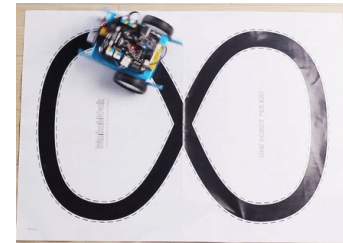
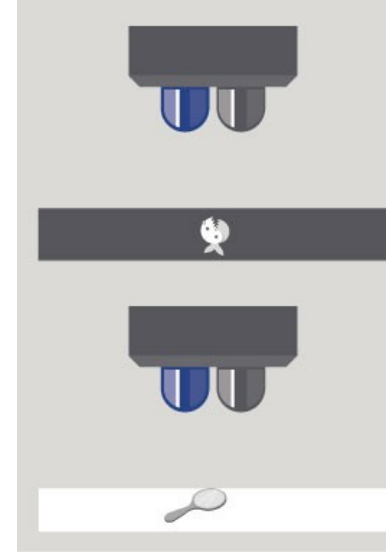
Actividad N°2: Hacer un contador de interrupciones

Aplicaciones de la compuerta óptica:

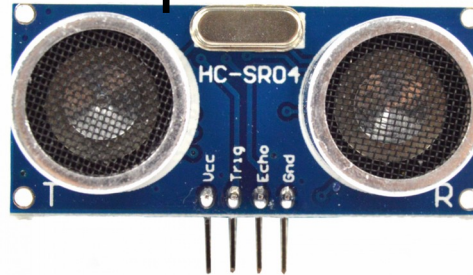
Velocidad Angular



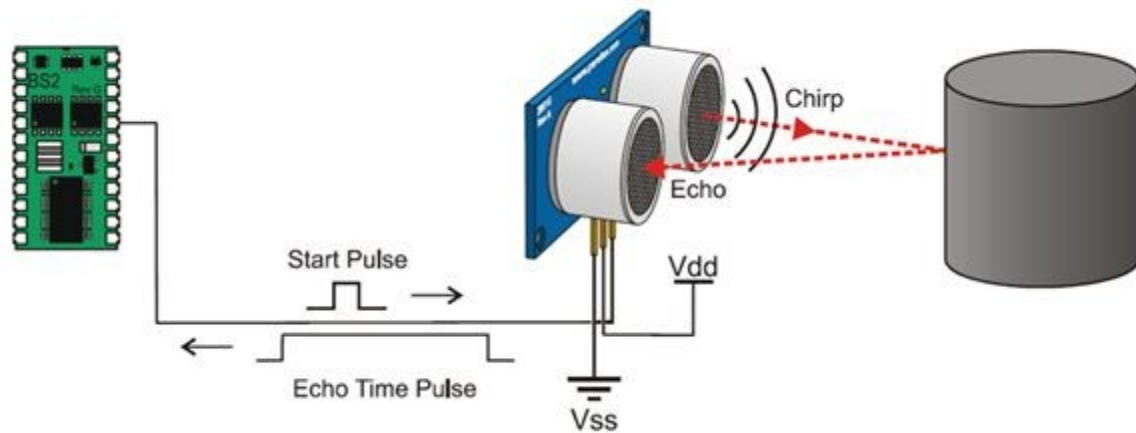
SEGUIDOR



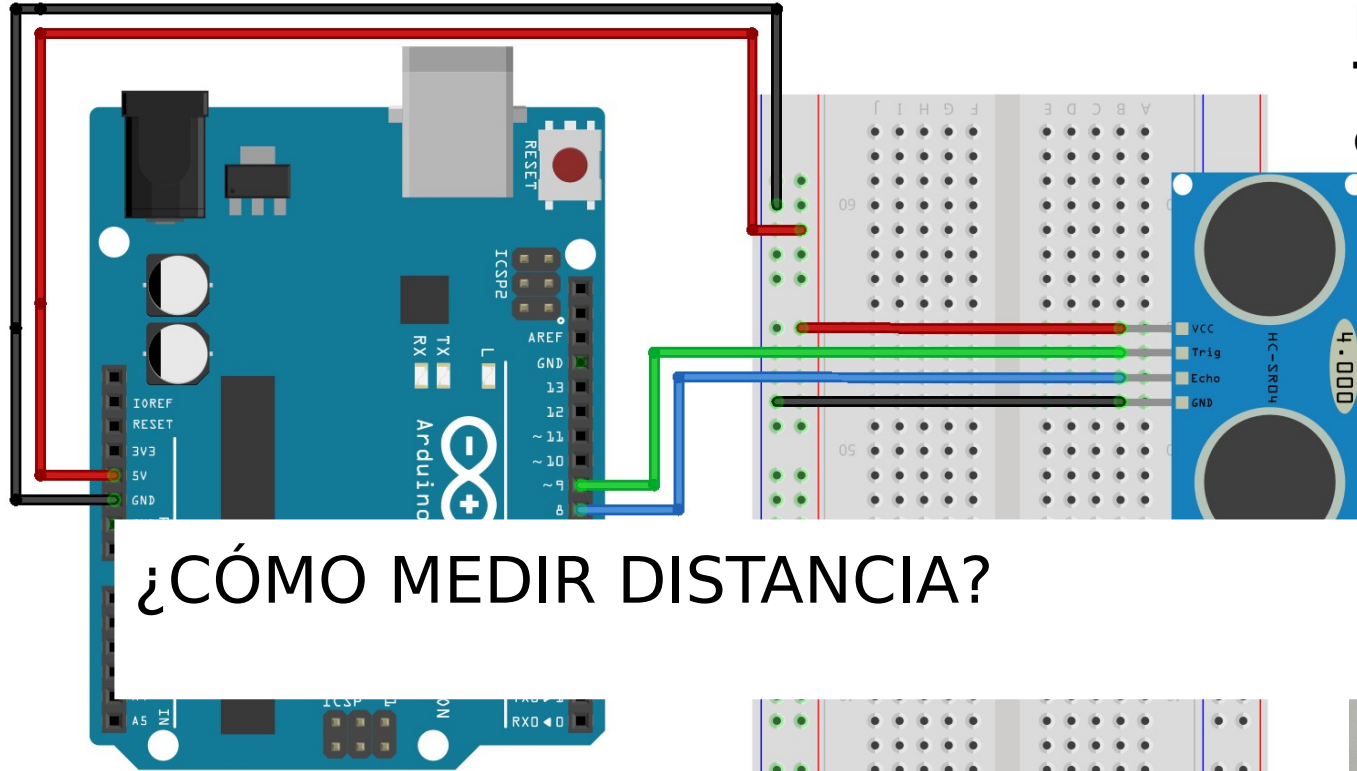
Sensor Ultrasónico: El sensor envía un sonido de 40kHz genera eleva el voltaje a 5V. Cuando recibe el eco, disminuye el voltaje a 0V, generando un pulso de igual ancho al tiempo que demora al sonido ir y volver. Si velocidad del sonido es 340m/s aprox., entonces podemos calcular la posición del objeto.



Acceder a [Hoja de Datos Sensor Ultrasonico HC-SR04](#)

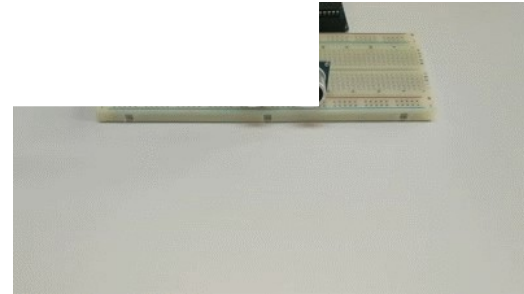
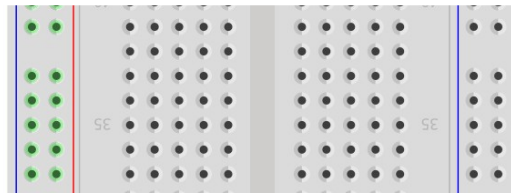


Conexión del Sensor Ultrasónico a Arduino



Ejecutar
TiempoUltraSonido.
o

¿CÓMO MEDIR DISTANCIA?



Presentación de Prácticas de Física

Conservación de la Energía y Cantidad de Movimiento

1. Conservación de la Cantidad de Movimiento. Función de Transferencia.
Sensor de velocidad
2. Trabajo de la Fricción. Coeficiente de roce dinámico. Sensor de Fuerza
3. Sensor de Fuerza
4. Medición del Momento de Inercia. Conservación de la Cantidad de Movimiento del eje. (Fotopuerta) Cantidad angular

Movimiento Armado

5. Periodo de un Péndulo Simple (Fotopuerta)

**Anotar Grupos
y Prácticas**